DOCUMENT RESUME

ED 178 264 SE 027 648

AUTHOR Larkin, Jill H.
TITLE Models of Strategy for Solving Physics Problems.
PUB DATE 16 Mar 79
NOTE 19p.; Paper presented at the annual meeting of the American Educational Research Association (San Francisco, California, April 8-12, 1979)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Cognitive Development; College Science; *Computer Assisted Instruction; Higher Education; Instructional Materials; Mental Development; *Models; *Physics; *Problem Solving; Science Education; Scientific Methodology; Scientific Principles; *Skill Development
IDENTIFIERS Equations (Mathematics)

ABSTRACT

A set of computer implemented models are presented which can assist in developing problem solving strategies. The three levels of expertise which are covered are beginners (those who have completed at least one university physics course), intermediates (university level physics majors in their third year of study), and professionals (university physics faculty). (Author/SA)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Jill H. Larkin

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED178264

Models of Strategy for Solving Physics Problems

Jill H. Larkin
Carnegie-Mellon University

16 March 1979

Draft of a paper for presentation at AERA session, Process Models of Skilled and Less Skilled Behavior in Technical Domains, San Francisco, April 8 - 12, 1979.

Session # 27.1

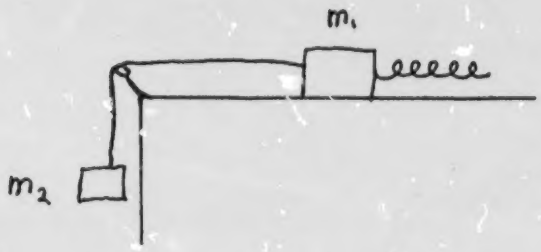
SE 027 648

This paper describes some work on building models of how people (including people of more and less skill) solve some problems in physics. The problems I address are paper-and-pencil, textbook problems in college-level mechanics, for example, like the one in Figure 1. Although very simple, problems like these do have something in common with real research problems in science: the solver is confronted with a physical situation (for example, blocks and pulleys and strings), and is asked to apply scientific theory in order to relate and predict data (here the acceleration of block 1). Thus problems like this seem to me to constitute a domain which is both sufficiently simple that there is some hope of understanding how expertise in solving them develops; but which still captures some of the intriguing richness of creative scientific problem solving.

The subjects I have studied solving such problems range in experience from university students, who had just completed their first one semester course in mechanics, to university professors who have taught such courses within the past few years. My experimental technique is to ask the subjects to read and solve physics problems "thinking aloud as much as possible." Ordinarily, after solving one practice problem, subjects can work in this way fairly comfortably (although there is no question that thinking aloud affects their work to a certain extent). The record of a problem solution then consists of both the typed transcript of the subject's "think-aloud protocol," together with anything he has written down on paper. This rich data is reduced by some sort of coding which extracts, according to well specified rules, key features for further analysis. For example, here I shall be talking about the order in which principles are applied, and about characteristic errors.

To account for the coded protocols, I am developing computer-implemented simulation models. A simulation model "works" in that it can be given (in stereotyped form) a problem like the one in Figure 1. The computer model solves the problem by outputting a series of steps which, can be compared with coded protocols from human solvers. The power of such a model depends on its ability to account accurately for the human data. In this paper I shall be talking in particular about how certain changes in the computer model can account for more and less skilful problem solving.

Figure 1: A sample textbook physics problem.



A spring having a spring constant k = 50 N/m is kept extended a uniform amount x = 0.3 m as it is used to pull the block of mass m_1 = 5 kg in the diagram above. This block moves without appreciable friction along the horizontal surface, and is connected by a light flexible cord running over a light frictionless pulley to a block of mass m_2 = 2 kg.

What is the acceleration of the block of mass m_1?

1. Work of a Human Solver

Figure 2 shows the work on paper of a moderately skilled human solver, who proceeded in the following steps.

- He sketched the situation, including the two blocks, cord, and string.
- For each of the two blocks, he sketched and labeled arrows indicating the forces acting on that block. On block 1, he shows the leftward tension force T due to the string, and the rightward force due to the spring, having a magnitude of k Δ x. For block 2 he shows the upward tension force T and the downward gravitational force $m_2 g$.
- He translated the information about forces into equations describing the acceleration of each block. The first equation says that the total force on block 2 is equal to $m_2 g - T$. This total force is equal to the mass of block 2 times its acceleration a. The second equation makes a similar statement about the total force F_1 on block 1.
- The remainder of his work, below the horizontal line in Figure 2, is simply algebra required to combine the two preceding equations to express a, the desired acceleration of block 1, in terms of quantities specified in the problem statement.

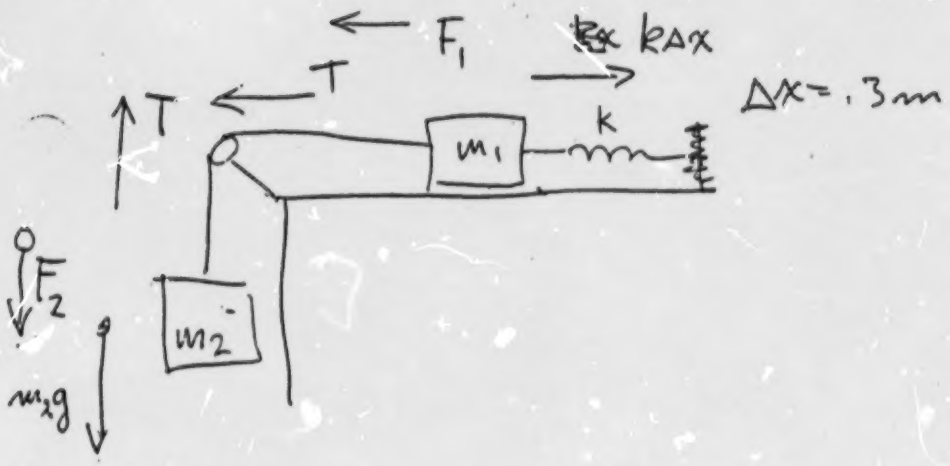
2. Output of the Model

In the interests of concreteness, I shall next describe the work of the model solver, i.e., the output of the computer program. I leave until later the discussion of how this model works.

Figures 3 and 4 show the work of the model (called PH100) on the problem in Figure 1. PH100 does not receive the problem in humans' language, but in the stereotyped form shown in Figure 3. Thus it is given a list specifying the desired quantity (a), the known quantities (m_1, m_2, k, and x), and a stereotyped sketch of the situation. This "sketch" is intended to represent, in computer-readable form, information which human solvers commonly abstract from the English problem statements and immediately write on their papers. Thus, like the human solver whose paper is shown in Figure 2, PH100 uses a sketch which indicates that block 2 is a hanging block with mass m_2 that it is connected by a string going over a pulley to block 1, which in turn lies on a horizontal smooth surface and has a mass m_1.

PH100's next step is to construct a physical representation for the problem, corresponding

Figure 2: Work of a human solver.



$$F_2 = m_2 g - T = m_2 a$$

$$F_1 = T - k\Delta x = m_1 a$$

$$m_2 g - k\Delta x = (m_1 + m_2) a$$

$$a = \frac{m_2 g - k\Delta x}{m_1 + m_2} = \frac{2(10)\,\frac{kg\text{-}m}{sec^2} - 50\,\frac{N}{m}(.3)m}{7\ kg}$$

$$= \frac{20 - 15}{7} = \boxed{\frac{5}{7}\ m/sec^2}$$

to the labeled force arrows in the human solver's work. This step is labeled PHY-REP in Figure 4. Like the human solver, PH100 indicates for each block, a label (e.g., F-SPRING) and direction for each force on that block.

Once this has been done, PH100 translates each of these physical representations into a basic equation (labeled BASEQNS in Figure 3). For example, the basic equation for block 2 states that the gravity and the tension forces must account for the acceleration of the corresponding mass. (PH100 does not actually work with equations, but with lists of the quantities appearing in an equation.)

From there on PH100, like the human solver, works to eliminate from its equations quantities which are neither known nor desired. These eliminations are accomplished by applying physical principles to generate expressions which can be substituted for undesired quantities. For example, the first step in the work of PH100 is to substitute for the force F-SPRING using the principle (PRIN-EQN) stating that F-SPRING is equal to an expression containing the spring constant k and the extension x.

When all possible substitutions have been made, PH100 combines the results for the two systems (blocks 1 and 2) to obtain the answer, an "equation" involving the desired quantity a, and the quantities m_1, m_2, k and x specified in the problem.

3. Mechanisms of the Model

I turn now to how PH100 works. PH100 is written in OPS2, a language for implementing production systems, a type of computer program used to simulate human performance for a variety of tasks (Newell&Simon, 1972, Klahr, D. & Wallace, J.G., 1976). Like all production systems, PH100 has a small, unstructured, short-term *working memory*, as well as a large, homogeneous, long-term *production memory*. In addition, PH100 has a *paper* memory on which it can write things it wishes to remember.

Working memory contains approximately 7 elements to which PH100 is currently attending. Such elements can include current goals, bits of information just read in from the paper,

Figure 3: Problem as given to PH100.

(a)

```
(SKETCH (NON)
        (BLOCK2 (STATE HANGING)
                (MASS (M2)))
        (STRING (STATE OVER-PULLEY))
        (BLOCK1 (STATE HOR-SURF SMOOTH)
                (SURF-ANGLE HOR)
                (MASS (M1))
                (MO-DIR RIGHT)
                (MO-CH INC))
        (SPRING (SPR-CONST (K))
                (EXTENSION (X)))
        (UNK))

(DESIRED (A))
(KNOWN (M1) (M2) (G) (K) (X)))
```

Figure 4: Work of PH100.

```
(PHY-REP (2 (RIGHT-FORCE TENSION) (LEFT-FORCE F-GRAV))
         (1 (RIGHT-FORCE F-SPRING) (LEFT-FORCE TENSION)))

(BASEQNS (2 (F-GRAV) (TENSION) (MASS) (A))
         (1 (TENSION) (F-SPRING) (MASS) (A)))

(WORK (EQN (NOT (SUBST (TENSION))) (SUBST (F-SPRING)) (MASS) (A))
      (PRIN-EQN (F-SPRING) (K) (X)))
(WORK (EQN (NOT (SUBST (TENSION))) (K) (X) (SUBST (MASS)) (A))
      (PRIN-EQN (MASS) (M1)))
(1 (EQN (NOT (SUBST (TENSION))) (K) (X) (M1) (A)))

(WORK (EQN (M2) (G) (NOT (SUBST (TENSION))) (SUBST (MASS)) (A))
      (PRIN-EQN (MASS) (M2)))
(WORK (EQN (SUBST (F-GRAV)) (TENSION) (MASS) (A))
      (PRIN-EQN (F-GRAV) (M2) (G)))
((2 (EQN (M2) (G) (NOT (SUBST (TENSION))) (A)))

(ANSWER (EQN (M2) (G) (A) (K) (X) (M1) (A))
```

intermediate results, or final results to be written down on the paper.

Each *production* in the long-term memory consists of a condition-action unit. The condition side includes of patterns which can be satisfied by elemerts in working memory. The action side includes actions to be taken whenever the specified elements are found in working memory. PH100 currently consists of about 100 such production units, which can be conceptualized as acting in parallel, watching working memory, and waiting for their conditions to be satisfied. When a production finds that all its conditions are satisfied, the corresponding action is implemented[1]. An action can add elements to working memory, or write something on the paper. The parallel nature of the production memory gives production systems a flexible responsiveness to the current situations, which makes them particularly appropriate for psychological modeling (Newell, 1973, McDermott, 1978).

PH100's paper memory, intended to be analogous to a human solver's paper-and-pencil work, is flexible and addressable. PH100 can write in this memory any significant result, and, if later this result would be useful, it can be retrieved. This process seems analogous to the way people use organized space on a paper to record key results.

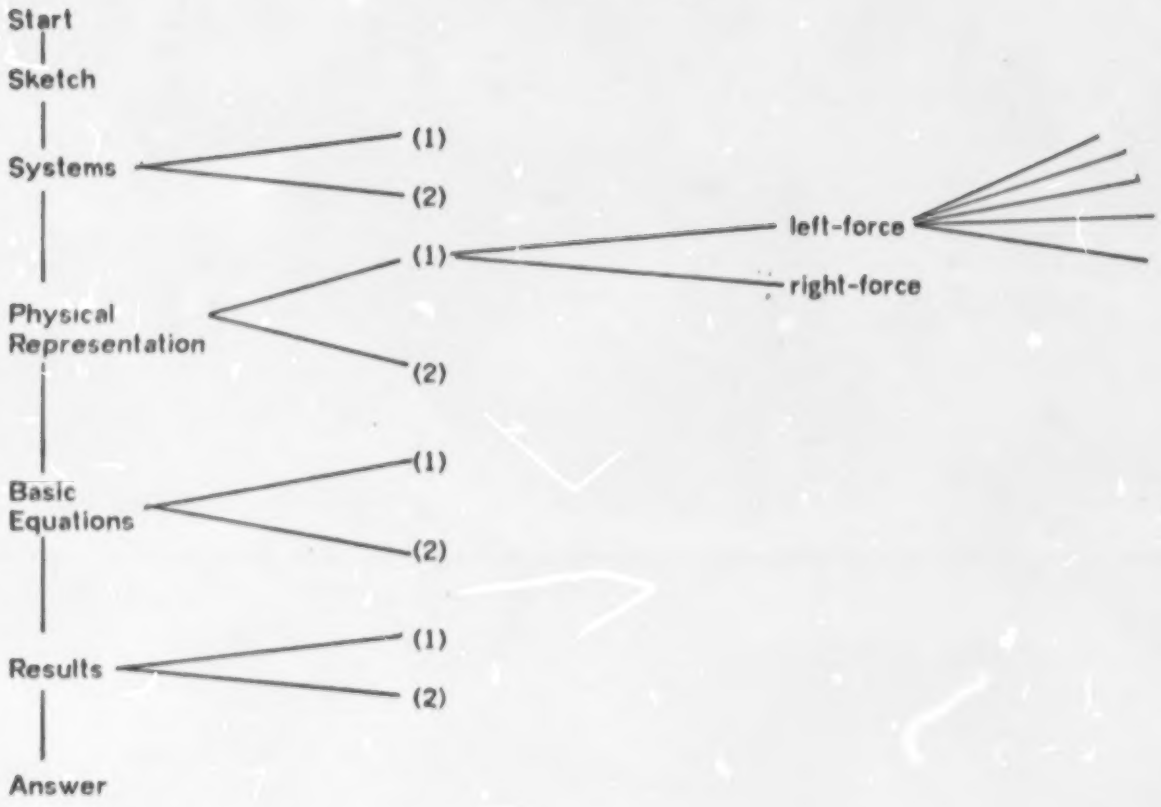
Figure 4 shows PH100's goal structure for the problem given in Figure 1. After studying the sketch, it works to construct a physical representation, finding for each system (1 & 2), left and right forces. Determining the nature of each of these forces may require considerable search of memory and processing of information. However, once the information is obtained, it is written on the paper, and cannot be lost again.

After finding left and right forces on a system (e.g., tension and F-spring on system 1), PH100 labels these forces by the name of the system (see Figure 4). Thereafter PH100 makes no effort to develop a physical representation for system 1, because the information on the paper specifies that this work has been completed.

---

[1] The situation is somewhat more complex because often more than one production is satisfied. For a discussion of the associated conflict resolution, see (Forgy&McDermott, 1976)

Figure 5: Goal Structure for PH100



The paper thus serves two crucial functions. It is a memory in which key results are stored, and it controls processing by specifying when certain work is complete.

In short, PH100 as described above, uses an information-development strategy. It begins with the sketch, showing known information, then develops from this sketch a qualitative representation of some theoretical objects like forces. This physical representation is then developed into equations which are ultimately developed into an answer.

4. A More Primitive Model

The information-development strategy described for PH100 is sophisticated in that it does not work directly toward the goal of finding an answer; but instead depends on strategic knowledge about how to develop existing information. A more primitive strategy would focus on the desired quantity (the acceleration of block 1) and work directly towards finding that. How could PH100 be modified to work in this more primitive way?

Figure 6 shows a goal structure for a more primitive program. The top goal is now always to get the answer. The first subgoal is to get some relevant equation, a subgoal accomplished by looking for the desired quantity (e.g., a), and finding a principle which involves that quantity (e.g., F = ma). This principle must then be instantiated (restated with values relevant to the current problem) so as to get a starting equation. Then the program turns to eliminating from this first equation quantities which are neither known nor desired. In this phase of its work it works exactly like the original PH100, locating quantities which are neither known nor desired, and applying physics principles to generate expressions which can be substituted for these values.

The basic difference between the two programs is the following. The original PH100 had a very sophisticated way of using a physical representation to generate a first basic equation or equations. Once this was done, the algebraic task of eliminating undesired quantities was fairly brief. The primitive version of the program, on the other hand, employs a very straightforward method for getting a first equation, and correspondingly the algebraic path

Figure 6: Goal structure of primitive PH100

Answer

Get equation

For desired, find principle

Instantiate

Find substitute

Find principle

Instantiate values

Combine --> new equation

from this equation to the final answer is likely to be long. Furthermore, because the primitive model does not generate key information as part of a physical representation of the problem, it encounters considerable difficulties in instantiating principles as discussed below.

5. Output of the Primitive Model

In discussing the output of the primitive model, I shall concurrently address how difficulties encountered by the model account for errors which are characteristic of inexperienced human solvers (first-year university students).

Figure 7 shows a summary of how the primitive PH100 solves the problem given in Figure 1. Because the desired quantity is a_1, acceleration of block 1, the first equation is the instantiated principle $F = ma_1$.

Since F is neither known nor desired PH100 seeks a principle which will allow substitution for F. The appropriate principle is that F is the total force on some system of objects. But no such system has ever been specified. Thus in order to function, PH100 must be provided with some rule which allows it find forces without a system. I have used a rule suggested a study of beginning physics students, which suggests that such students generate forces corresponding to "active" objects, in this case the gravitational force of the earth and the force due to the spring (Lochhead & Clement, 1978). This result is then combined with the original equation to obtain the equation $F_{sp} - F_g = ma_1$.

The next task is to substitute for F_{sp}. Part of this task is easy, using the well-known principle that the magnitude of this force is equal to the spring constant k times the extension x. However, this principle is usually stated in the form $F_{sp} = -kx$, leading PH100 to complete uncertainty about the appropriate sign for this quantity. This difficulty never arises with the more powerful version of PH100, which specifies its direction in its physical representation. Among human solvers, 5 out of 11 beginners expressed uncertainty about the sign of this force, while none of the 10 experienced solvers expressed any uncertainty.

The next task is to substitute for the gravitational force F_g, using the principle that F_g

Figure 7: Work of primitive PH100.

Equation (desired a_1): $F = m\,a_1$

Substitute: F

Principle: Find forces

Instantiate: $F = F_{sp} - F_g$

Combine: $F_{sp} - F_g = m\,a_1$

Substitute: $F_{sp} = kx$

Substitute: $F_g = mg$ $(m = m_2)$

Combine: $kx - m_2 g = m\,a_1$

Substitute: $m = m_1 + m_2$ or $m = m_1$

equals mass m times the gravitational constant g. However, since a system was never specified, what mass is to be used? The appropriate mass is m_2, the mass of the hanging block 2, but the uncertaintly inherent in the primitive model was reflected by 1 of the 11 beginning solvers who used the total mass $m_1 + m_2$. No skilled solver expressed any confusion.

The final task is to substitute for the mass m, appearing with the acceleration a_1. Again no system has been specified so PHI00 must in some way find this mass. A simple way is just to use the mass of the object associated with the desired acceleration, i.e., m_1. This is not, however, a decision leading to a correct result. If the forces considered are the spring force and the gravitational force, then the system accelerated by these forces is the combined system involving m_1, m_2, and the string. Thus the appropriate mass is $m_1 + m_2$. Of the 11 beginning solvers, 5 inappropriately used the mass m_1 as suggested by the primitive model, while at most 1 experienced solver made this error.

6. Order of Principle Application

Clearly the primitive version of the PHI00 predicts a rather different order in which principles will be applied than the more powerful version. For the problem in Figure 1, the powerful model predicts that the first principles will be those elucidating the directions of forces on individual blocks, followed by application of F = ma to account for the acceleration of each block. The primitive model predicts that the first principle will be $F = ma_2$ applied to block 1, followed by attempts to find active forces. Similar contrasting predictions are made for the several problems PHI00 can solve.

By observing the order in which principles are applied, each human protocol can be classified as corresponding to primitive or powerful version of the model. The results of such classification for three problems are shown in Figure 8. The order in which experienced solvers apply principles reflects preferentially the more powerful model, while the beginners' work corresponds more often to the primitive model.

Figure 8: Classification of problem solutions.

Problem 2

	Powerful	Primitive
Experts	9	1
Beginners	2	9

Problems 1 and 4

	Powerful	Primitive
Experts	20	0
Beginners	7	12 (3 couldn't be classified)

7. Summary

In summary, PH100 is a running computer model, which solves some simple paper-and-pencil mechanics problems. In its primitive mode, PH100 seems to account for some key features shown preferentially by inexperienced solvers. In its more powerful mode it seems to account for similar features in the work of more experienced solvers.

The major difference between these two versions of PH100 are: In its powerful version, PH100 focuses on developing knowledge, starting with a sketch and developing more and more information about the problem until the answer is obtained. In its primitive version, PH100 focuses on obtaining the answer, continually setting subgoals directly necessary for obtaining that goal.

Because it is always working toward the answer, the primitive version of PH100 is automatically quantitative in emphasis. It is always trying to generate an appropriate equation, and addresses qualitative issues (e.g. which way forces point) only when forced to do so. In contrast, the powerful version of PH100 first develops a qualitative representation of the problem and only subsequently develops this representation into quantitative equations.

In the powerful version of the model, the physics (e.g., how forces account for motion) is prominent and used early in the problem solution. In the primitive version, algebraic knowledge is prominent, and physics knowledge is only used when necessary to implement this information.

References

McDermott, J. & Forgy, C. *Production system conflict resolution strategies.* Technical Report, Department of Computer Science, Carnegie-Mellon University, 1976.

Klahr, D. & Wallace, J.G. *Cognivitve development: An information processing view.* : Lawrence Erlbaum Associates 1976.

Lochhead, J. & Clement, J.(eds.). *Cognitive-Process Instruction.* : The Franklin Institute Press, Philadelphia 1978.

McDermott, J. Some strengths of production system architectures. In *NATO A.S.I. proceedings: Structural/process theories of complex human behavior,* : A.W. Sijthoff, International Publishing Company, The Netherlands, 1978.

Newell, A. Production systems: Models of control structures. In Chase, W.G. (Ed.), *Visual information processing,* : Academic Press, New York, 1973.

Newell, A. & Simon, H.A. *Human Problem Solving.* : Prentice-Hall, Inc. 1972.